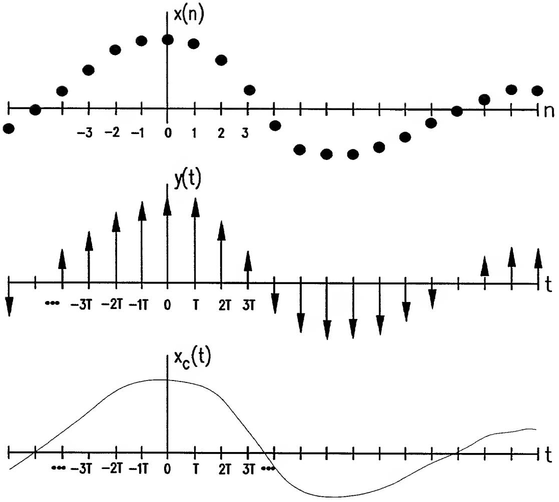


FIG. 1

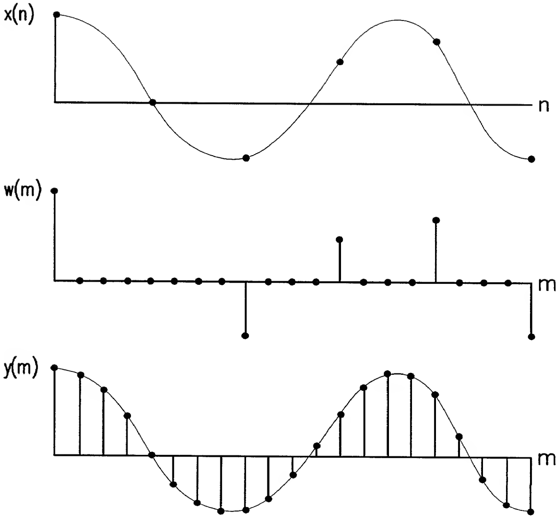


FIG. 2

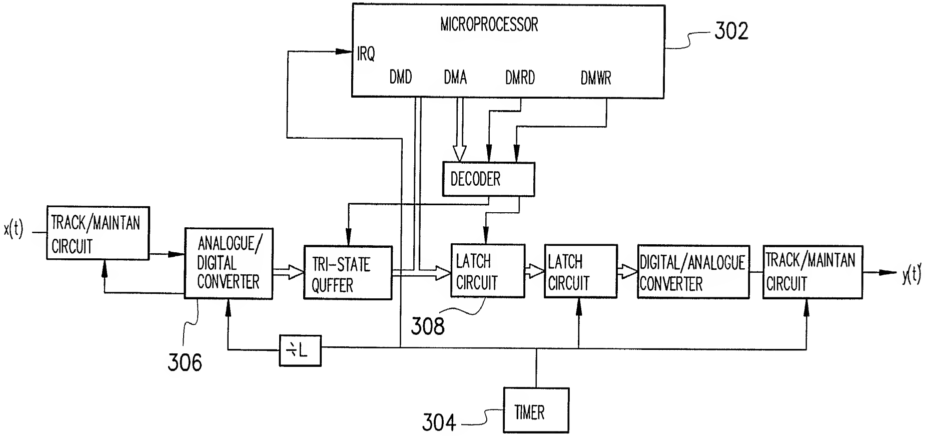
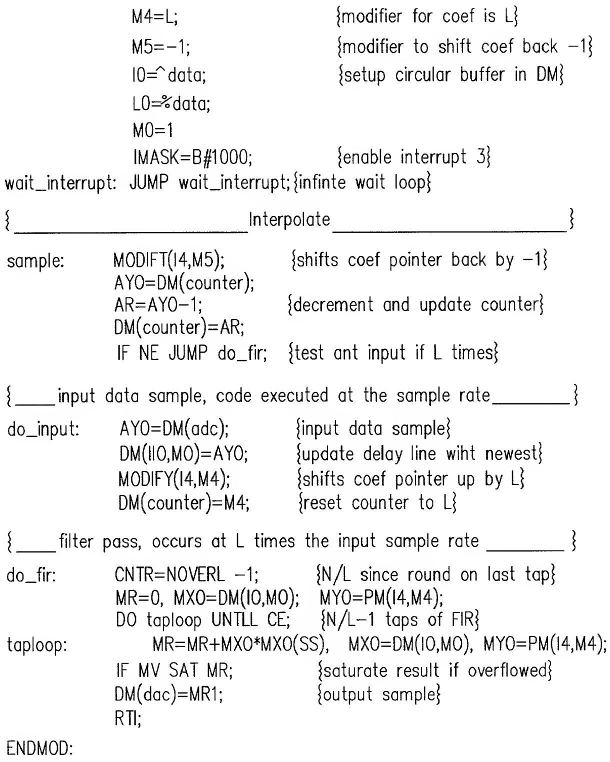

MICROPROCESSOR
302
IRQ
DMD
DMA
DMRD
DMWR
DECODER
x(t)
TRACK/MAINTAN CIRCUIT
ANALOGUE/ DIGITAL CONVERTER
TRI-STATE QUFFER
LATCH CIRCUIT
LATCH CIRCUIT
DIGITAL/ANALOGUE CONVERTER
TRACK/MAINTAN CIRCUIT
y(t)
308
306
÷L
TIMER
304

FIG. 3

```
{ INTERPOLATE.dsp
Real time Direct Form Filter, N taps, uses an efficient algorthm
to interpolate by L for an increase of L times the input sample rate. A
restriction on the number of taps that N/L be integer.

    INPUT: adc
    OUTPUT: dac

}

MODULE/RAM/ABS=0 interpolate;
.CONST              N=300;
.CONST              L=4:                 { interpolate by factor of L }
.CONST              NoverL=75;
.VAR/PM/RAM/CIRC    coef[N];
.VAR/PM/RAM/CIRC    data[NoverL];
.VAR/PM/RAM/        counter;
.PORT               adc;
.PORT               adc;
.INIT               coef:<coef.dat>;


          RTI;                          {interrupt 0 }
          RTI;                          {interrupt 1 }
          RTI;                          {interrupt 2 }
          JUMP sample;                  {interrupt 3 at (L*input rate) }

initialize:         IMASK=b#0000;       {disable all interrupts}
                    ICNTL=b#01111;      {edge sensitive interrupts}
                    SI=1;               {set interpolate counter to 1}
                    DM(counter0=SI;     {for first data sample}
                    I4=^coef;           {setup a circular buffer in PM}
                    L4=%coef;
```

(listing contiunes on nest page)

FIG. 4



```
                 M4=L;                   {modifier for coef is L}
                 M5=-1;                  {modifier to shift coef back -1}
                 I0=^data;               {setup circular buffer in DM}
                 L0=%data;
                 M0=1
                 IMASK=B#1000;           {enable interrupt 3}
wait_interrupt:  JUMP wait_interrupt;{infinte wait loop}

{_________________________Interpolate_________________________}

sample:        MODIFT(I4,M5);        {shifts coef pointer back by -1}
               AY0=DM(counter);
               AR=AY0-1;             {decrement and update counter}
               DM(counter)=AR;
               IF NE JUMP do_fir;    {test ant input if L times}

{____input data sample, code executed at the sample rate________}

do_input:      AY0=DM(adc);          {input data sample}
               DM(I10,M0)=AY0;       {update delay line wiht newest}
               MODIFY(I4,M4);        {shifts coef pointer up by L}
               DM(counter)=M4;       {reset counter to L}

{____filter pass, occurs at L times the input sample rate ________}

do_fir:        CNTR=NOVERL -1;         {N/L since round on last tap}
               MR=0, MX0=DM(I0,M0);    MY0=PM(I4,M4);
               DO taploop UNTLL CE;    {N/L-1 taps of FIR}
taploop:           MR=MR+MX0*MX0(SS),  MX0=DM(I0,M0), MY0=PM(I4,M4);
               IF MV SAT MR;           {saturate result if overflowed}
               DM(dac)=MR1;            {output sample}
               RTI;

ENDMOD:
```

FIG. 4

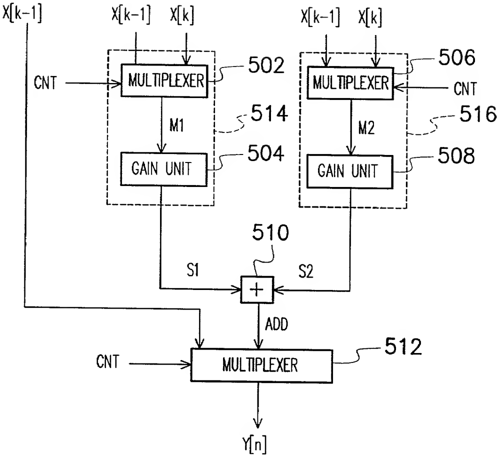

X[k−1]
X[k−1]
X[k]
X[k−1]
X[k]
CNT
MULTIPLEXER
502
506
MULTIPLEXER
CNT
514
M1
516
M2
504
GAIN UNIT
508
GAIN UNIT
510
S1
S2
+
ADD
CNT
MULTIPLEXER
512
Y[n]

FIG. 5

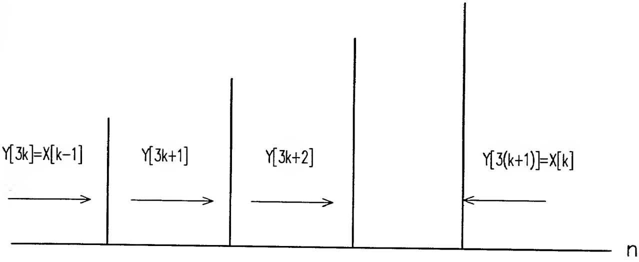

Y[3k]=X[k−1]
Y[3k+1]
Y[3k+2]
Y[3(k+1)]=X[k]
n

FIG. 6